



Procedia Computer Science

Volume 29, 2014, Pages 1915–1925

ICCS 2014. 14th International Conference on Computational Science



A Web-Based System for Visualisation-Driven Interactive Multi-Objective Optimisation

Jan Hettenhausen¹, Andrew Lewis¹, and Timoleon Kipouros²

¹ Griffith University, Brisbane, Queensland, Australia
j.hettenhausen@griffith.edu.au, a.lewis@griffith.edu.au

² University of Cambridge, Cambridge, United Kingdom
tk291@eng.cam.ac.uk

Abstract

Interactive Multi-Objective Optimisation is an increasing field of evolutionary and swarm intelligence-based algorithms. By involving a human decision, a set of relevant non-dominated points can often be acquired at significantly lower computational costs than with *a posteriori* algorithms. A rarely addressed issue in interactive optimisation is the design of efficient user interfaces and the application of interactive optimisation as a design tool in engineering applications. This paper presents advances in the design of user interaction and user interfaces in combination with interactive multi-objective particle swarm optimisation algorithm. The case study used for this research is an actively investigated aeronautics engineering problem.

Keywords: Interactive optimisation, user interface design, multi-objective optimisation, engineering design

1 Introduction

In real-world applications optimisation problems often require simultaneous optimisation against several conflicting objectives. In general, no global optimum exists that satisfies all relevant objectives. Instead, objectively, the solution to such a problem presents itself as a set of points which cannot be further improved in any of their objectives without a detrimental effect on at least one other objective. Without loss of generality, a multi-objective optimisation problem (MOP) can be stated as:

$$\begin{aligned} \text{minimise } \vec{f}(\vec{x}) &= \{f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x})\} \\ f_{k \in [1, m]} : \mathbb{R}^n &\rightarrow \mathbb{R}, \forall \vec{x} \in \mathcal{S} \subseteq \mathbb{R}^n \end{aligned} \quad (1)$$

Parameter vectors $\vec{x} \in \mathbb{R}^n$ are subject to boundary constraints forming the feasible space \mathcal{S} , commonly denoted *parameter space*. Its image, subject to \vec{f} , is the *objective space*.

A decision vector \vec{x}_1 is said to dominate another decision vector \vec{x}_2 , denoted $\vec{x}_1 \prec \vec{x}_2$, iff $\vec{f}(x_1, i) \leq \vec{f}(x_2, i), \forall i \in 1, \dots, m$ and $\exists i \in 1, \dots, m : \vec{f}(x_1, i) < \vec{f}(x_2, i)$. The set of globally non-dominated decision vectors is the Pareto-optimal set, its counterpart in objective space is the Pareto-front.

The dominance relation represents a partial order of the decision vectors. Equal importance of all objectives assumed, vectors in the Pareto-optimal therefore represent the formal solution to the MOP. However, in practical applications only one or, at most, a small number of solutions are required. Commonly this is achieved by invoking a human decision maker (DM) to provide their subjective preferences. Depending on when these preferences are captured in relation to the search process, multi-objective optimisation methods can be categorised as *a priori*, *a posteriori* or interactive algorithms, respectively.

In *a priori* methods invocation of the decision maker occurs prior to beginning the optimisation process. While effective methods for modelling and integrating *a priori* preferences into the search exists, approaches in practice often suffer from the difficulty of accurately articulating preferences for a yet unknown problem domain. The effectiveness of some methods can be further reduced when applied to problems with concave Pareto-fronts.

In contrast, in *a posteriori* approaches preferences are not utilised until after the search is completed, commonly in a separate post-analysis step. Algorithms of this class attempt to find a comprehensive approximation to the entire Pareto-front. *A posteriori* methods rank among the most popular for a broad range of MOPs. However, their computational cost is generally considered to be higher than methods of the other classes and the additional cost and effort for an effective post-analysis is often disregarded. Their computational cost for problems with four or more objectives is often considered to be prohibitive.

In interactive methods preferences are captured and refined progressively during the optimisation process. This puts the DM in a position to build an understanding of the topology of the problem domain while gradually directing the search towards regions of interest. Interactive methods generally perform favourably in terms of finding desirable results at computational costs lower than by a comparable *a posteriori* approach. The more focussed search also allows more effective handling of problems with four or more objectives, commonly called many-objective problems, where most *a posteriori* methods drop significantly in effectiveness. A limitation of interactive approaches is their requirement for human interaction at multiple times during the optimisation process.

The ability for the decision maker to gain a better understanding of the problem, as well as the reduced computational cost and handling of larger numbers of objectives have triggered a growing interest in interactive methods in recent years. An indication of this trend can be observed in the reviews by Jaszkiwicz and Branke [16] and Purshouse *et al.* [23]. However, the focus largely lies on the integration of Multi-Criteria Decision Making (MCDM) with only a small proportion of work focussing on other ways of user interaction.

The comprehensive work available in MCDM provides an attractive and well supported foundation for decision making. A limitation of MCDM is that the structured nature of decision capturing, to an extent, dictates the way the user has to interact with the system based on algorithmic necessities rather than user requirements. While the performance of user interfaces is hard to evaluate, human fatigue is a frequently discussed issue in the literature.

A different approach to interaction based on user-centric visual interfaces was identified in, e.g. [12, 23]. However, methods of this category are still extremely rare. The concept behind this is to provide information to the user in a way they can handle efficiently and integrate decision making in an intuitive, non-obstructive way. Emphasis is placed on facilitating a better understanding of the decision space by the user and reduced fatigue through more targeted

interaction.

In a previous study, the authors have developed such a system based on multi-objective particle swarm optimisation (MOPSO)[12]. The prototype featured a static user interface using parallel coordinates [14] for the visualisation of non-dominated candidate solutions and a scatter-plot to allow plotting any two parameters or objectives against each other. In addition, each candidate solution, in this case an airfoil shape, could be selected and graphically inspected. Results from this work compared favourably to a contemporary *a posteriori* approach.

Given the success of the approach, a future path for this kind of interaction was developed in collaboration with aeronautics engineers and a range of improvement to the user interface and the method of interaction were developed. In this paper the prototype for the resulting new user interface and the improved interaction methodology are presented.

The remainder of this paper has the following structure. Related work on interactive optimisation and visualisation will be discussed in section 2. Advances to the user interface and the method of interaction will be presented in Section 3 and 3 respectively. An outline of the underlying architecture and implementation is given in 4. Summarising remarks and a brief outlook on future work can be found in Sections 5 and 6.

2 Related Work

Interactive optimisation was originally developed in Operations Research. Within Operations Research, decision making formally developed into the field of Multi-Criteria Decision Making (MCDM). Optimisation techniques using meta-heuristics, e.g. based on Evolutionary Algorithms or Swarm Intelligence, largely focussed on *a posteriori* and, to a lesser extent, *a priori* approaches, with interactive methods representing a comparatively small niche [23, 9, 24, 26].

A majority of publications primarily focussed on tackling the algorithmic integration of MCDM with evolutionary algorithms or, to a lesser degree, other meta-heuristics. Capturing of preferences is usually performed by asking the DM to perform pairwise comparisons, rank solutions or specify goal vectors in objective space. Preference models, e.g. in the form of utility functions, are then constructed based on these preferences and used to augment the dominance relation accordingly. Such approaches include the works by Deb and Kumar [6], Deb *et al* [7], Sinha *et al* [25] and Yadav *et al* [27]. A method for discrete optimisation was presented by Phelps and Köksalan [21]. A non-evolutionary algorithm-based approach was developed by Agraval *et al.* [1] integrating a utility function Multi-Objective Particle Swarm Optimisation. Extensive reviews of the current literature can be found in the reviews by Jaszkievicz and Branke [16] and Purshouse *et al.* [23].

User interfaces for the majority of approaches are limited to showing plots or numerical data for selected points with user interface design generally limited to capturing information required for the respective MCDM approach. A more user-interface centric approach was developed by Fonseca and Fleming [11] with further developments by Fleming *et al.* [10]. The multi-objective genetic algorithm (MOGA) presented captures aspirations and, in later versions priorities, from the decision maker. A parallel coordinates chart of the objective space is presented to the DM from which a desired solution can be selected and used as goal information.

A user interface for various MCDM-based methods was developed by Deb and Chaudhuri [5]. Their application, termed I-MODE, uses scatterplots to visualise the objective space for problems with two or three objectives. Preference information can be captured for use with either the weighted sum approach, a utility function based approach, Chebyshev function-based methods or trade-off information. The use of scatterplots currently limits the applicability to problems with more than three objectives.

A commonality between these approaches is limitation of visual representations to the objective space. This stands in partial contrast to the goal of enabling the decision maker to increase their understanding of the problem and further rules out the possibility for the decision maker to form preferences based on parameter values. While outside the scope of common MCDM methods, such preferences can have practical value in, for example, engineering applications, where computational models cannot accurately capture all aspects of manufacturability and durability of design that a design expert can identify [18, 17].

In contrast to this, post analysis techniques, as used in *a posteriori* methods, frequently aim to present this data to the decision maker allowing selection based on domain expertise and requirements. Data representation is usually done using sophisticated visualisation tools to facilitate visual analysis. Examples of such post analysis tools are the Heatmap visualisation by Pryke *et al.* [22] and the parallel coordinates-based works by Kipouros *et al.* [20, 19], based in parts on the works by Inselberg [14, 15].

An initial approach to extend Heatmap visualisation to a user interface for interactive optimisation was developed by Hettenhausen *et al.* [12]. The approach allowed the decision maker to explore known solutions, including dominated ones, via a dynamically adjustable heatmap. Points could be selected as preferred guide particles for a MOPSO algorithm. While the method allowed focussing on a region of interest with good accuracy, a shortcoming of the approach was the high frequency of interactions required.

The authors' previous approach [13] was based on an even stronger focus on techniques used in post analysis. The method captures preferences in the form of preferred ranges in parameter space, relying on human expertise to determine desirable solutions and guiding a MOPSO algorithm accordingly. The user interface was primarily visual, using parallel coordinates to present parameters and objectives of known non-dominated solutions. In addition, the decision maker can select any two axes to visualise in a scatterplot. A visual inspection of rendered candidate solutions could be opened by clicking on a solution in either plot, which also highlights the point in the other visualisation modes. In addition to using preferences as guidance information, the underlying algorithm also employs virtual guide particles used to drive the search to selected regions with currently no, or only few, solutions.

Work on this method has continued in collaboration with aeronautics engineers and using a real-world optimisation problem. The outcome of this, regarding user interface design and method of interaction, will be discussed in the following sections.

3 User Interface for Interactive Optimisation

The user interface developed as part of Interactive MOPSO in Hettenhausen *et al.* [13] provided a way for the decision maker to employ three linked visualisations to analyse the known, non-dominated solutions and articulate their preferences in the form of constraints in parameter space. In this first iteration, a static layout with fixed components was used, limiting the user to one scatterplot and a view of one selected candidate solution, in addition to the Parallel Coordinates chart. Using the airfoil case-study, this user interface was tested and evaluated in collaboration with aeronautics engineers and a detailed plan for improvements to the user interface, the method of interaction and the algorithm was developed. The work presented in this paper represents the advances made to the user interface and the method of interaction based on this evaluation. Inspiration was also taken from established post-analysis tools, as, for example, used by Kipouros *et al.* [20, 19].

To achieve more flexibility, the static layout of the application was removed and replaced with a newly created dynamic “Desktop” like environment, making it possible to move GUI

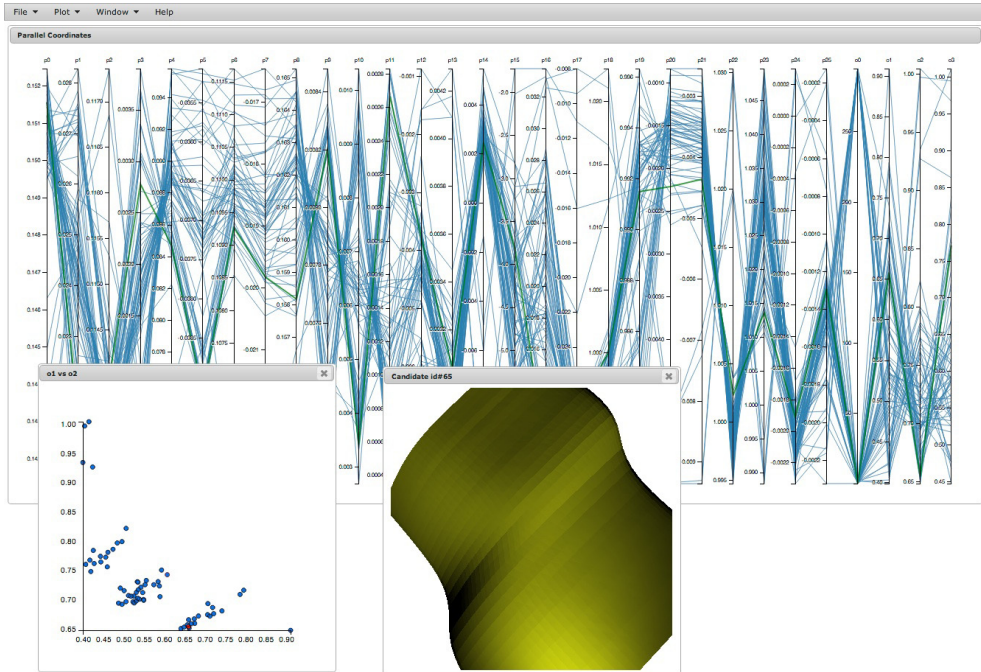


Figure 1: User Interface Overview. The selected solution is highlighted in green in the parallel coordinates plot and red in the scatterplot. The third window shows a 3D model of the candidate solution.

elements around freely. It also makes it possible to create and remove graphs dynamically based on requirements. A screenshot of the new environment is shown in Figure 1. The core principal of the user interface, i.e. the idea to let the user interact via a primary graphical representation of the entire space of available solutions, concretely the parallel coordinates plot, has remained structurally similar.

Despite this conceptual similarity, a range of changes have been made to how the visualisation can be used. Graphs can now be dynamically created and removed, allowing multiple scatter-plots of different axes to be shown at the same time. Additional linking makes it possible to select individual points or ranges in any of the plots, with selections in the other plots appearing accordingly. In the same way, multiple points can be visualised as 3D models at the same time, allowing comparison of two or more concrete candidate designs. At present this visualisation is still limited to the physical model. However, an extension integrating flow visualisations and other performance characteristics is under development.

As common in visualising results from multi-objective optimisation, the set of solutions shown in plots is limited to the currently known non-dominated set by default. To observe the progression of a search and more easily detect some connections between parameters and objectives, the decision maker can add historic solutions to the visualisation as well.

While most of these features can be found in various post analysis tools, most of their functionality has, to the authors' knowledge, not been applied to ongoing analysis and decision maker in interactive optimisation.

sectionInteractive Optimisation with MOPSO The user interface was built to facilitate dis-

covery of the problem and decision making in a flexible manner. In this section core elements of the workflow for interactive MOPSO will be discussed. In summary, the DM can communicate preferences by either creating boundary constraints on parameters where desired, or set individual parameters to fixed values or set desirable ranges on one or more of the objectives. Additional flexibility was also added to the virtual guide particle feature introduced in [12]. Virtual guide particles allow the algorithm to try and expand the range of parameter values by creating virtual guides rather than using archived solutions. Virtual guides are constructed using a combination of known good parameter values and points generated based on ranges provided by the DM. Their development was motivated by fact that in search spaces with sparsely distributed valid decision vectors effective guidance of the swarm and expansion into desirable search areas of the search space becomes increasingly difficult. While a decision may have a clear idea for a desirable range for one or several parameters, constructing a valid parameter vector based on this idea is rarely feasible. Targeted selection of desirable guide particles, on the other hand, can be a possible solution but, for common heuristics, may not successfully lead to an expansion of the search space if the infeasible area is sufficiently large. At what point this occurs depends on the diversity of known good points, the sparsity of the space and the mechanics by which the algorithm combines known good points to produce new candidate solutions. In the case of PSO, new candidates are generated from linear combinations of known good solutions. A constant factor combined with a constrained random variable are used to limit and vary the influence of the chosen guide particles. Virtual guide particles provide an elegant addition in that they integrate with the algorithm without changing its mechanics while at the same time adding a means to bias the search towards a specific region of the search space without requiring knowledge of an actually feasible point in this area. This method proved to be effective in overcoming patches of infeasible space and directing the search to desirable regions for varying numbers of parameters. To allow more control over this feature, additional controls allow the DM more fine-grained control over the frequency and extent of virtual guide particles, values previously set as static probabilities.

Initialisation After an experiment is created from scratch, i.e. based on a population of randomly generated parameter vectors, interaction is generally not beneficial until a reasonable population of non-dominated points is found. While this represents a common case, the decision maker is also given the option to start with a full or partial population generated by another experiment or simply based on known good points acquired by other means. In this case, interactions may be useful from the beginning. In the case of a new start, prior knowledge may be applied immediately. However, practical experiments have shown that limiting diversity of the population very early on can have a detrimental effect to the progress due to the sparsity of valid configurations in parameter space.

User Interaction The sparsity of publications addressing the user interface side of interactive optimisation and aspects usability suggests that this problem is a major challenge in the field. Consequently, few real world applications appear to employ interactive optimisation. The user interaction presented here draws its justification from empirical but widely established post analysis techniques commonly used in combination with *a posteriori* methods in engineering and other disciplines, notably the focus on visual representations and the interaction with the parameter space stem from these approaches.

In a typical interaction, the decision maker will be presented with the parallel coordinates representation of the known non-dominated set. A benefit of this representation is that high dimensional spaces can be represented in an intuitive and simple way. Common implementations

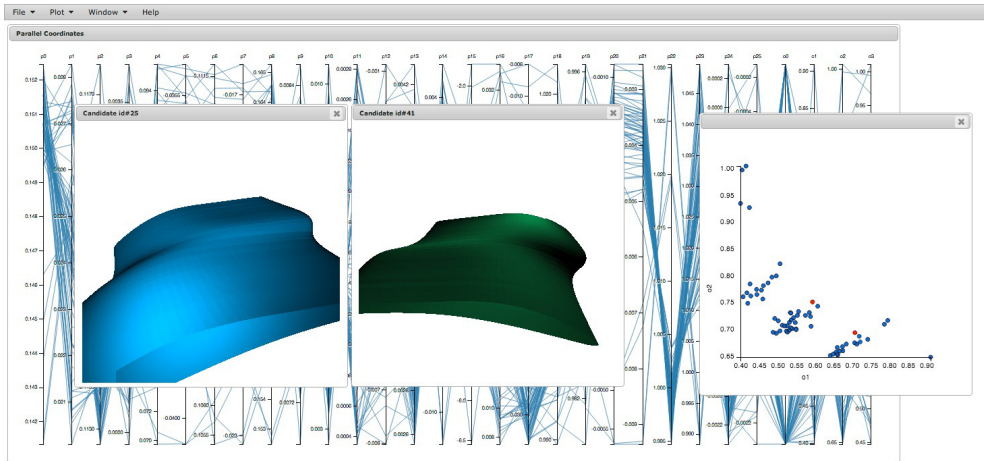


Figure 2: The scatterplot of objective three and four is used to compare designs. The selected points are highlighted in the scatterplot.

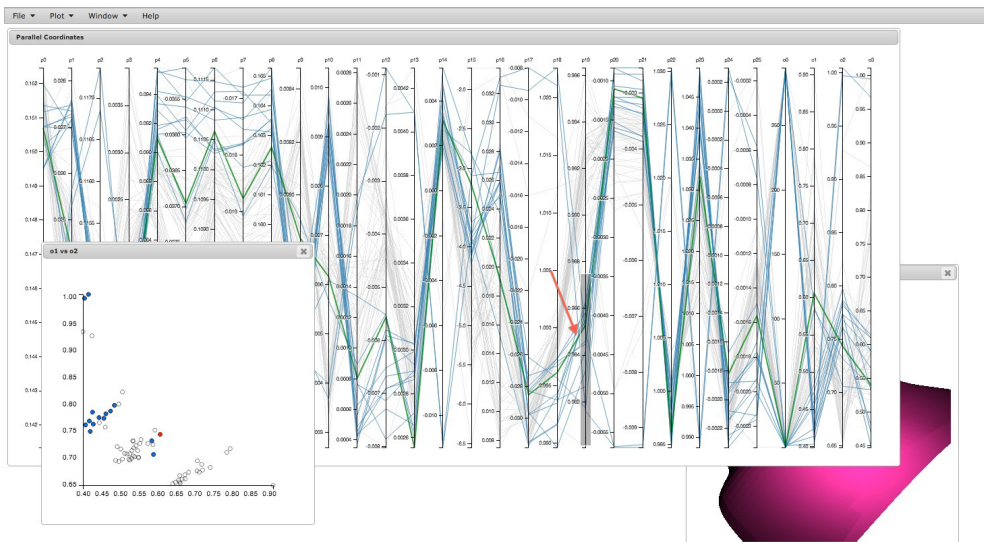


Figure 3: Parallel Coordinates and Scatter plot are used to analyse the relation of Parameter 19 to objective 2 and 3 and mark a range.

of parallel coordinates further include the possibility to select ranges for solutions and remove or visually distinguish between solutions that match the selected range(s).

In this interactive method, the decision maker can use sliders to select, analyse and discover patterns in the data and to select ranges based on domain expertise or because they correlate strongly with solutions deemed desirable. Different parameters will correlate to such solutions to varying degrees, ranging from no correlation at all, to a clear link between a parameter

value and a certain type of candidate point. The option to also show solutions from previous iterations in the graph can aide the decision maker in finding patterns in the data, particularly when the set of known non-dominated solutions is small.

Due to its capability of visualising the entire space in one graph, parallel coordinates was chosen as the primary interface for the user to interact with and manipulate the data. Scatterplots are available as a supplementary representation of the data. The decision maker can add any number of scatterplots specifying dimensions to be plotted against each other per plot. This representation allows further analysis of correlations between two dimensions and allows, for example, highlighting links between two parameters the DM might find interesting, or showing the relation between a parameter and an objective. Selections in one scatterplot are also highlighted in all other plots, including the parallel coordinates and may be used to articulate preferences. The capability to dynamically create additional graphs was added to enable the decision maker to perform side by side comparisons of correlations between dimensions, or of multiple individual solutions.

To evaluate individual candidate solutions more effectively, an interactive 3D plot can be opened for each point. An example showing both plots and two selected candidate solutions is shown in Figure 2.

Instead of fixed intervals for decision making, updates to preferences can be made at any time during the search and updated preferences are used from that point onwards. In practical experiments, fixed intervals were often perceived as a limitation, as sometimes a decision maker had to wait several iterations to make a change based on a new observation.

For parameter values identified as desirable, an appropriate range can then be selected that will be used as a constraint by the algorithm. This can either be done to limit a parameter to a small range or to set a lower or upper limit and allow exploration in the open direction. Ranges can be specified far beyond known solutions to facilitate exploration in that direction (see Figure 3). If such a range is specified, the algorithm will start generating virtual guide particles in this area. The frequency of random guide particles being used for new candidate points can be adjusted depending on how much emphasis the decision maker wants to set on furthering expansion of the search into this area. On the other hand, the possibility to fix parameters to specific values was added, effectively removing the parameter from the search and reducing dimensionality of the problem. This step can also be found in other post analysis processes.

Empirically this form of interaction generally yields good results. However, in some cases the DM may choose to simply set a goal direction in one or more objectives. In response, the possibility to select ranges on objectives was added and the algorithm was modified to take both types of preferences into account when choosing guide particles and creating virtual guide particles. An adaptive model to more accurately handle this combination is currently under development.

Finishing an Experiment Other than for *a priori* and *a posteriori* approaches, explicit stopping conditions are generally not needed for interactive optimisation, with the exception of comparative studies where an equivalent budget of iterations or function evaluations is given to the competing methods. Instead the decision maker can freely choose to end the search when a satisfactory outcome has been achieved or no further progress is made. An addition in the framework presented is the option to revert to a previous iteration to start another search from that point with different preferences. This caters to the possibility that more than one area of the search space is of interest to the decision maker.

4 System Architecture

Engineering design has increasingly evolved into a team process, across institutions and organisations. Consequently, the ability to easily access software and data from multiple sites and collaborate and share results and experiments was an important design aspect of the system.

The user interface was implemented using current web-technologies, allowing users access via a standard compliant web-browser. The web-interface is driven by a Python-based web-application serving experiment data and models as well as forwarding user input. A middle-ware was developed in Python to store and process experiment data, manage interactive MOPSO instances, in the following denoted as MOPSO, and interact with an HPC cluster.

In this architecture the web-application only provides a thin layer for handling input and output requirements of the user-interface. The business logic is primarily implemented in the middle-ware and made available via an XML-RPC-based API. This API allows retrieval of experiment data and models for the 3D visualisation as well as functionality to create and control experiments in the form of instances of MOPSO. Currently a Portable Batch System (PBS)-based HPC cluster is used to evaluate candidate solutions. Job scheduling, monitoring and result retrieval is handled by the middle-ware and is transparent to MOPSO and the user. Communication with the cluster is handled via SSH with each candidate design being submitted to the PBS job scheduler as a separate job. To ensure reliability and the ability to resume the experiment, the state of each running instance of MOPSO and of all cluster jobs is stored in a persistent database. This allows the system to recover from restarts, loss of connectivity and lost HPC jobs in most situations.

All requirements for the user interface and visualisations could be satisfied with current web-technologies, in particular SVG [4] and WebGL [8] in combination with the OpenSource software libraries D3 [2] and ThreeJS [3].

The user interaction with MOPSO is currently designed for either a single decision maker or, if desired, a group of decision makers agreeing on one set of preferences. The possibility to synchronise selections to allow such group decision making from multiple sites is currently being considered as a future extension. Collaboration focusses on the ability to make experiments available to other users to discuss results and to create new experiments based on the final, or any intermediate historic state of any experiment. An experiment workflow can therefore be to create a new experiment starting with a random population or pick up an interesting state of another experiment and progress with different preferences. In addition, all results and historic states can be downloaded as CSV (comma separated values) or JSON (JavaScript Object Notation) files or analysed using the visualisations provided by the web-interface.

An added benefit of a web-application from a development point of view is the possibility to continuously roll out improvements to the system while requiring only minimal maintenance intervals as changes can be made in one central location. Such future improvements will include the possibility to add solution visualisations, such as the currently used 3D model, via a plugin API, additional features for collaboration, metadata handling and additional algorithms for interactive optimisation.

5 Conclusion

A user interface for interactive multi-objective optimisation with MOPSO was presented. A range of improvements to the user interface, as well as to the interaction between the user and the algorithm, were discussed in the context of the current literature on interactive optimisation as well as the previous work the authors have done with the underlying algorithm and a prior

web-based environment for interaction. Parallel advances to the underlying algorithm, e.g. regarding the changed handling of preferences, are currently under investigation using a real-world compressor blade optimisation problem for an aeronautics application. Results of the algorithm performance and objective comparisons will be discussed in a separate publication. The work presented in this paper is the result of collaboration with users of the system. Aspects of usability were established empirically with users of the system and can be further supported based on the use of established methods and techniques from post analyses.

6 Future Work

Future work on the project will focus on the refinement of the user interaction and improving the accuracy of capturing guidance information from the DM. Aside from more fine-grained controls, additional visualisation types, such as, for example, heatmap visualisation are conceivable. Algorithmic development will be conducted in parallel.

On the implementation side, emphasis will be put on developing the current prototype into a production ready system, allowing optimisation of arbitrary MOPs. Foremost this will include a plugin architecture, as outlined in section 4.

References

- [1] S. Agrawal, Y. Dashora, M.K. Tiwari, and Y.-J. Son. Interactive particle swarm: A pareto-adaptive metaheuristic to multiobjective optimization. *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, 38(2):258–277, March 2008.
- [2] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D3: Data-driven documents. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011.
- [3] Ricardo Cabello et al. three.js - javascript 3d library.
- [4] Erik Dahlström, Patrick Dengler, Anthony Grasso, C Liley, Cameron McCormack, Doug Schepers, and Jonathan Watt. Scalable vector graphics (svg) 1.1. *World Wide Web Consortium Recommendation*, 16, 2011.
- [5] K. Deb and S. Chaudhuri. I-MODE: An interactive multi-objective optimization and decision-making using evolutionary methods. In *Evolutionary Multi-Criterion Optimization*, pages 788–802. Springer, 2007.
- [6] Kalyanmoy Deb and Abhishek Kumar. Interactive evolutionary multi-objective optimization and decision-making using reference direction method. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 781–788, New York, NY, USA, 2007. ACM.
- [7] Kalyanmoy Deb, Ankur Sinha, Pekka J Korhonen, and Jyrki Wallenius. An interactive evolutionary multi-objective optimization method based on progressively approximated value functions. *Evolutionary Computation, IEEE Transactions on*, 14(5):723–739, 2010.
- [8] Chris Marrin (Editor). WebGL specification.
- [9] José Rui Figueira, Salvatore Greco, Vincent Mousseau, and Roman Słowiński. Multiobjective optimization. In Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Słowiński, editors, *Interactive and Evolutionary Approaches*, chapter Interactive Multiobjective Optimization Using a Set of Additive Value Functions, pages 97–119. Springer-Verlag, Berlin, Heidelberg, 2008.
- [10] Peter J Fleming, Robin C Purshouse, and Robert J Lygoe. Many-objective optimization: An engineering design perspective. In *Evolutionary multi-criterion optimization*, pages 14–32. Springer, 2005.

- [11] C.M. Fonseca, P.J. Fleming, et al. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the fifth international conference on genetic algorithms*, pages 416–423. Citeseer, 1993.
- [12] J. Hettenhausen, A. Lewis, and S. Mostaghim. Interactive multi-objective particle swarm optimization with heatmap-visualization-based user interface. *Engineering Optimization*, 42(2):119–139, February 2010.
- [13] Jan Hettenhausen, Andrew Lewis, Marcus Randall, and Timoleon Kipouros. Interactive multi-objective particle swarm optimisation using decision space interaction. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 3411–3418. IEEE, 2013.
- [14] Alfred Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985.
- [15] Alfred Inselberg. *Parallel coordinates: visual multidimensional geometry and its applications*, volume 20. Springer, 2009.
- [16] Andrzej Jaszkiewicz and Jürgen Branke. Interactive multiobjective evolutionary algorithms. In *Multiobjective Optimization*, pages 179–193. Springer, 2008.
- [17] R. Kamalian, H. Takagi, and A.M. Agogino. The Role Of Constraints and Human Interaction in Evolving MEMS Designs: Microresonator Case Study. *Proceedings of DETC*, 4, 2004.
- [18] Raffi Kamalian, Hideyuki Takagi, and Alice M. Agogino. Optimized Design of MEMS by Evolutionary Multi-objective Optimization with Interactive Evolutionary Computation. In *GECCO-2004*, pages 1030–1041, 2004.
- [19] Timoleon Kipouros, Alfred Inselberg, Geoffrey T Parks, and A Mark Savill. Parallel coordinates in computational engineering design. 2013.
- [20] Timoleon Kipouros, M Mleczko, and AM Savill. Use of parallel coordinates for post-analyses of multi-objective aerodynamic design optimisation in turbomachinery. *AIAA Paper*, 2138, 2008.
- [21] Selcen (Pamuk) Phelps and Murat Köksalan. An interactive evolutionary metaheuristic for multiobjective combinatorial optimization. *Management Science*, 49(12):pp. 1726–1738, 2003.
- [22] Andy Pryke, Sanaz Mostaghim, and Alireza Nazemi. Heatmap Visualisation of Population Based Multi Objective Algorithms. In *Evolutionary Multi-Criterion Optimization conference (EMO 2007)*, Japan, March 2007. Springer Verlag.
- [23] Robin C Purshouse, Kalyanmoy Deb, Maszatul M Mansor, Sanaz Mostaghim, and Rui Wang. A review of hybrid evolutionary multiple criteria decision making methods. Technical report, Computational Optimization and Innovation (COIN) Laboratory, Michigan State University, 2014.
- [24] Ankur Sinha. *Progressively Interactive Evolutionary Multiobjective Optimization*. PhD thesis, Aalto University, School of Economics, Department of Business Technology, 2011.
- [25] Ankur Sinha, Kalyanmoy Deb, Pekka Korhonen, and Jyrki Wallenius. Progressively interactive evolutionary multi-objective optimization method using generalized polynomial value functions. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010.
- [26] Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Kalyanmoy Deb. An improved progressively interactive evolutionary multi-objective optimization algorithm with a fixed budget of decision maker calls. Technical report, Aalto University, School of Economics, Department of Information and Service Economy, 2011.
- [27] S. R. Yadav, Yogesh Dashora, Ravi Shankar, Felix. T. S. Chan, and M. K. Tiwari. An interactive particle swarm optimisation for selecting a product family and designing its supply chain. *Int. J. Comput. Appl. Technol.*, 31(3/4):168–186, 2008.